

What's Open Liberty and JMS guide

Open liberty: Open source project that provides a flexible server framework for running and building java application built by IBM.

Jakarta Messaging Guide: Step-by-Step guide on Jakarta Messaging APIs to build the application and implement a messaging solution that enables communication between different parts of Java microservices.

Milestones

Initial Set up

- Set up development environment
- Fork the initial repository and contributed through PRs
- Configure the necessary tools such as maven, IBM MQ

Implementation

- Developed the **system** and **inventory** microservices
- Created the message producers and consumers (sendMessage(), onMessage())

Documentation

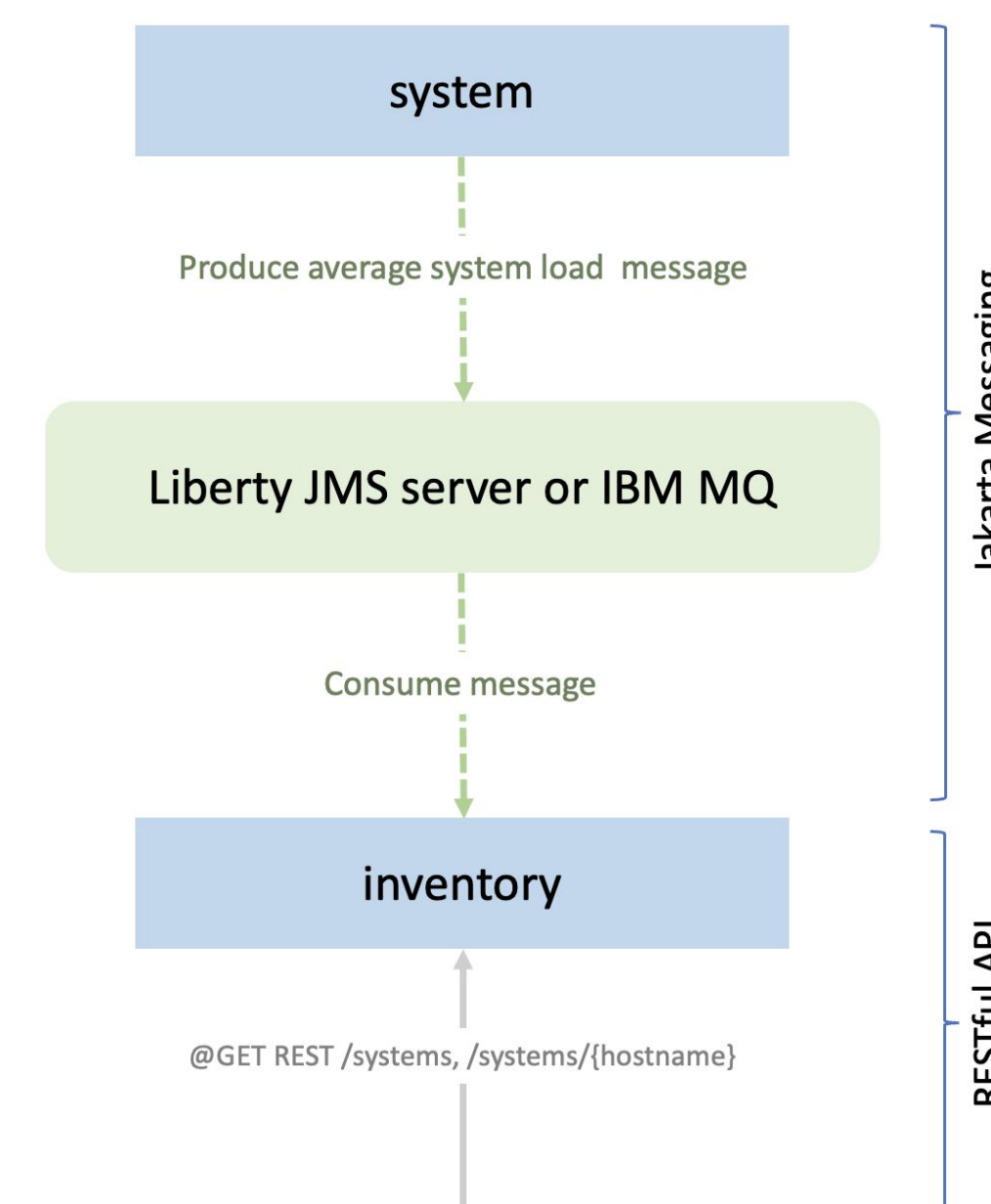
- Wrote comprehensive step-by-step guides (README.adoc)
- Provided an overview of setting up **Jakarta Messaging** and the roles of **message producers** and **consumers** in the application

Testing

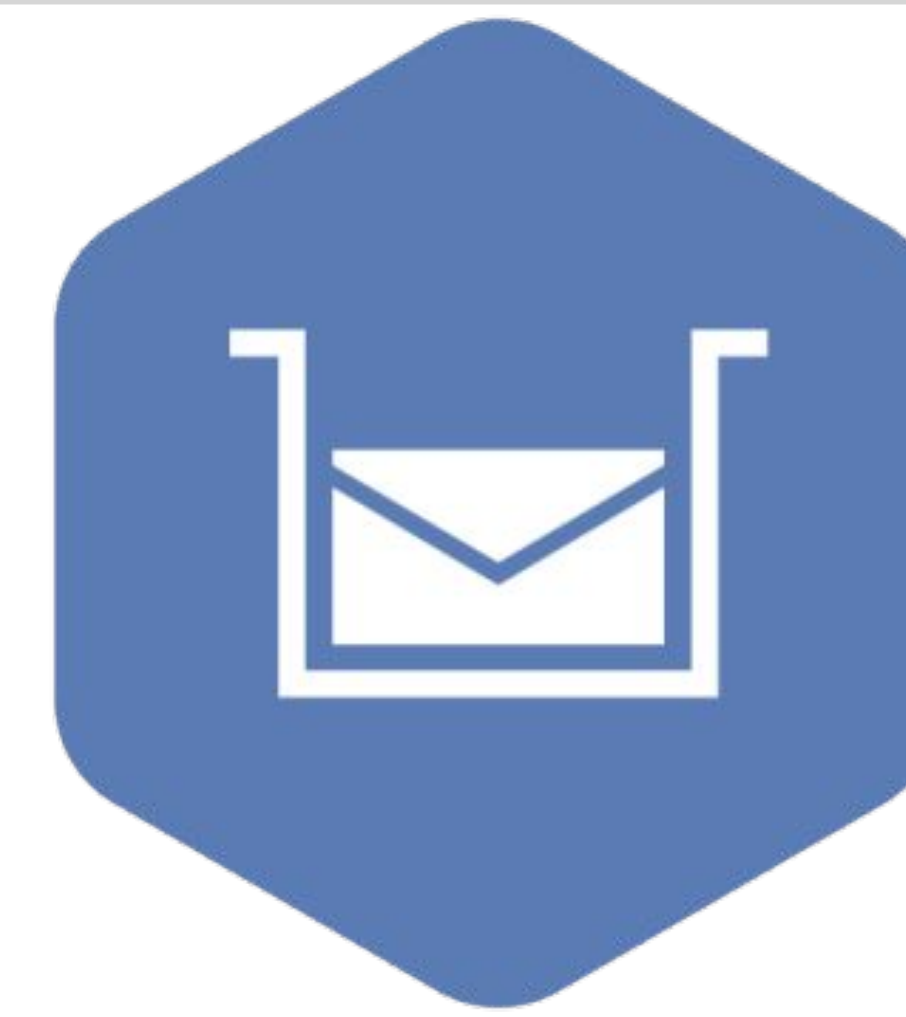
- Implemented **Integration Test** Class to ensure the reliability and performance of the application.
- Implemented **test.sh** for automated testing

Highlights and Accomplishments

Application Architecture



IBM MQ integration



- Designed the application container also to use **IBM MQ** as the messaging server., increasing both flexibility and compatibility.

- How to communicate between system and inventory services using **JMS server or IBM MQ**

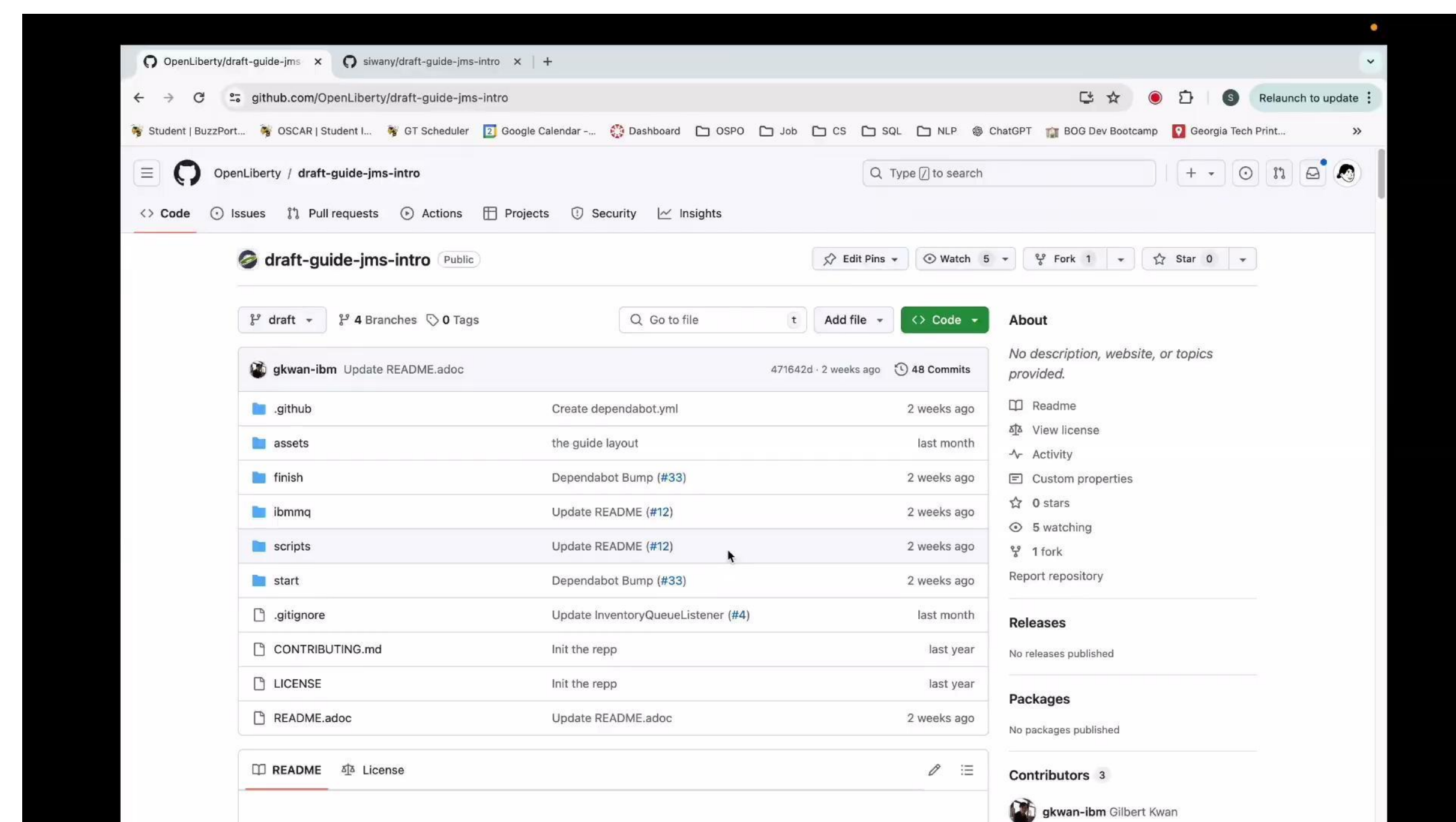
```
[INFO] [INFO] {"hostname":"Siwans-MacBook-Pro.local","loadAverage":5.06640625}
[INFO] [INFO] {"hostname":"Siwans-MacBook-Pro.local","loadAverage":5.068359375}
```

- When both **system** and **inventory** services are running, the application shows hostname and CPU System Load every 15 seconds

Github Link for the JMS guide

<https://github.com/OpenLiberty/draft-guide-jms-intro>

Application Demo Video



Engaging with Open Liberty

If you got interested in the Open Liberty project, here are some resources and contacts you can reach out to:



Open Liberty Guides

<https://openliberty.io/guides/>

If you want to be a contributor:

<https://openliberty.io/contribute/>

What Skills Did I Learn?

Web application development

- Designed and Implemented the web application by using JMS service, RESTful APIs, Maven

IBM MQ

- Integrated IBM MQ as a messaging server for message handling.

Technical Writing

- Wrote comprehensive step-by-step guides by providing detailed explanations

Automated Testing

- Developed junit integration tests to verify application functionality



Students: Pooja Saji
Mentors: Gilbert Kwan

What's Open Liberty?

Open Liberty is a lightweight, cloud-native application framework supporting MicroProfile and Jakarta EE APIs, designed for creating microservices.

Milestones

Milestone 1

- Learned several open liberty guides to understand certain developer tools like Maven and Gradle
- Contributed PRs to several test classes by updating them

Milestone 2

- Successfully set up a docker environment to run the local Open Liberty website
- Updated commands in the Deepdive Grade Guide and improved the README documentation to reflect these changes

Milestone 3

- Verified all of the commands in the Deepdive Grade Guide and the JMS Intro Guide
- Completed the peer review process for the JMS Intro Guide

Highlights and Accomplishments

Docker Environment

```
(base) poojasaji@Poojas-MBP Dock % docker run -d --name lgdev -p 4000:4000 --rm \
-v $(pwd)/content/guides:/devSite/openliberty.io/src/main/content/guides \
-v $(pwd)/content/img/guide:/devSite/openliberty.io/src/main/content/img/guide \
-v $(pwd)/content/blogs:/devSite/openliberty.io/src/main/content/_posts \
-v $(pwd)/content/blogs:/devSite/openliberty.io/src/main/content/_i18n/en/_posts \
-v $(pwd)/content/img/blog:/devSite/openliberty.io/src/main/content/img/blog \
lgdev:1.0
ae244a7c9c020ea600e6b8fc9b48e61a64d1f778621ba40d5d580c009a0a5ff4
(base) poojasaji@Poojas-MBP Dock % docker logs lgdev
Configuration file: src/main/content/_config.yml
Configuration file: src/main/content/_dev_config.yml
[
  Source: src/main/content
  Destination: /devSite/openliberty.io/_site
Incremental build: disabled. Enable with --incremental
Generating...
```

Set up the local docker environment
using lgdev commands to run the local
Open Liberty Website

Updating Deepdive Gradle Guide

Retrieve the PostgreSQL container IP address by running the following command:

```
podman inspect -f "{{.NetworkSettings.IPAddress }}" postgres-container
```

The command returns the PostgreSQL container IP address:

Updated all Docker commands to Podman commands and
adjusted the README to reflect the changes

Engaging with Open Liberty

Interested in Open Liberty?
Here are the following links

Open Liberty Guides

<https://openliberty.io/guides/>

Become a Contributor

<https://openliberty.io/contribute/>

Skills Learned

- Learned about developer tools such as MicroProfile, Jakarta EE, Maven, and Gradle through Open Liberty guides
- Learned how to contribute to open source projects using Github through both browser and command line using a GPG commit signature
- Learned how to collaborate in a work environment through daily scrums
- Learned how to provide feedback through peer review and test code